Smart Water Bottle with Nutritional Concentration Measurements (SWAN-USA)

Dominic Falvo, Farris Thrower, Gonzalo Gomez-Silva and Rhami Thrower

Dept. of Electrical Engineering and Computer Science, University of Central Florida, Orlando, Florida, 32816-2450

Abstract — The Smart Water Bottle with Nutritional Concentration Measurements (SWAN – USA) was designed to promote healthier lifestyles. The bottle functions similarly to other smart bottles found on the market. The bottle is equipped with an ultrasonic sensor that will inform users on how much water is in the bottle, an LED alert system to remind users to drink water, and a touch sensor to ensure the bottle has ample use time. What sets this bottle apart from other smart water bottles is that there will be a fixed wavelength spectrophotometer attached to the bottle. With some chemistry the spectrometer can inform users on how much sugar, in grams, is in the beverage. Our main goal in designing this bottle was to allow users to make more self-conscious decisions when drinking beverages.

Index Terms — Absorption, full stack, housing design, sensor, spectroscopy, ultrasonic applications.

I. Introduction

Water is vital to life. As such it is important to know how much water you are drinking to avoid dehydration. This is the simple concept that sparked the idea for S.W.A.N.-U.S.A. What if a smart water bottle would measure how much sugar is in the drink you are consuming? What if it could help verify the nutritional facts put on the label of beverages you like to consume? This is the goal of our smart water bottle.

Using spectroscopy and a suite of other sensors the smart water bottle will automatically track certain data points. Smart water bottles are not a new concept in the consumer market. There are already several products that track water intake. However, none deploy the use of optical systems to measure sugar concentrations. We will discuss the principles that lead us to implement this type of technology in later sections. One thing our team has kept in mind is that even though the components of the water bottle may be complicated the user interface must remain simple. This is to enhance the users' experience and attract a wider audience. This project will showcase aspects of three different disciplines: Electrical Engineering, Photonics Engineering, and Computer Science.

II. Specifications

This section will highlight the key component and system specifications for our system. The highlighted specifications in the table will show what we prioritized in the design of the smart water bottle. The specifications were chosen based on system performance, cost, and what accuracy of our readings.

Figure 1 outlines the specifications for our electrical engineering and system specifications. The highlighted parameters for the accuracy of ultrasonic sensor readings is the most important specification for our bottles functionality. Other parameters such as power efficiency of components and battery life are also outlined and will be discussed later in the paper.

Figure 2 shows the specifications of the optical sub system. Key characteristics for the optical set up is the signal to noise ratio of the sensor, and the accuracy in concentration measurements. Other parameters like the focal length of the lens, series resistance for the LED, and the optical output power in watts is also showcased in the table.

	component	parameter	specification
Electrical	Battery	life	≥ 3hrs
	Linear regulator	efficiency	≥80%
	Ultrasonic sensor	accuracy	≥80%
	Bluetooth	response	≤ 50 ms
	Device	compatibility	Android
	Bottle	Size	12”h x 6”d
	Data	Transfer time	<5 seconds

Fig. 1. Electrical Component Specification Table

	component	parameter	specification
Optical	Lenses	Focal length	≤40mm
	LED	Optical power	2 mW
	LED	Rs	55Ω
	Light sensor	SNR	≥40dB
	Sugar concentration	accuracy	≥80%

Fig. 2. Optical Component Specification Table

III. General Hardware Design

This section will show readers the hardware block diagram, and 3D modeling of electronics housing, optical system housing and overview of what the final product should look like. Figure 3 is the hardware block diagram and is a simple representation of how each part will function simultaneously. The hardware block diagram will also show the workflow distribution between team members.

The key components featured in our block diagram include the brain of the system, our microcontroller, our power supply design, optical subsystem with all its components, and our electronic peripherals.

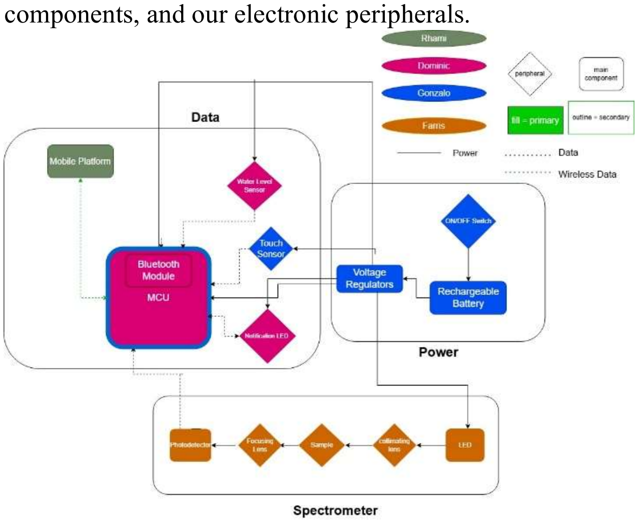

Fig. 3. Hardware block Diagram for SWAN-USA

A. *Overall Bottle Design*

This section shows 3D renderings of the entire bottle design. The following sections will break the design down into further detail, but this section is used to get a general idea of what the bottle will look like. The figure below is an isometric projection of the bottle. Some key features in the sketch include the opening in the lid for the ultrasonic sensor and the square base that will hold our hardware components.

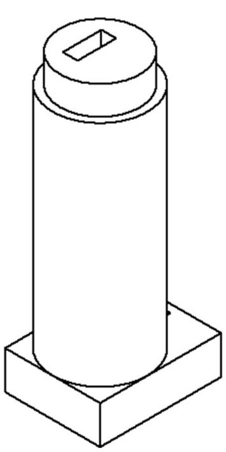

Fig. 4. 3D sketch of entire bottle design

The 2D image below highlights the key position of the cuvette, electrical housing, and the embedded sensor in the lid.

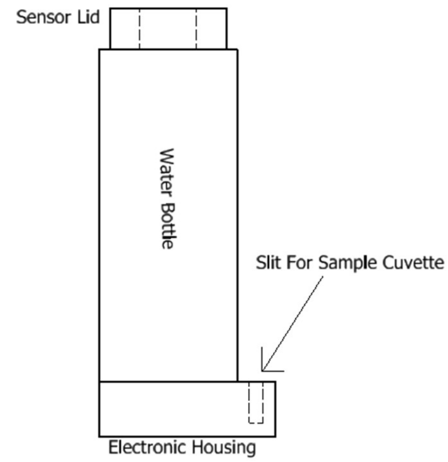

Fig. 5. 2D View of Bottle Design with Labels

B. *The Spectrometer Housing*

The design of the spectrometer required a bit more attention to detail. The position of the lens mounts, the sensor and the LED had to be precise. Any misalignment in the sub system would be catastrophic to the readings we get from the light sensor. We printed the housing to cut costs, because traditional amounts from various optical companies would double the cost of the system. The first figure in this section is a top view of the system that will show the spacing between components. The second image will depict the lid used to block out ambient light from the system.

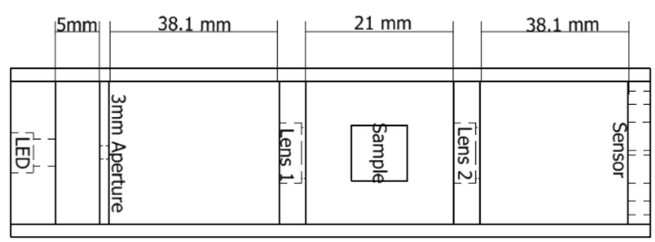

Fig. 6. Topological View of Spectrometer Sub System

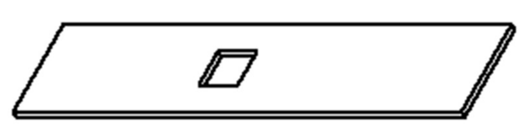

Fig. 7. Spectrometer sub system Lid

C. *Electrical Housing*

The electronics housing is made up of two parts. The first is a platform that will hold the PCB, power regulator, and battery pack. The second is a hollow cube that will be attached at the base of the bottle. The idea is the cube will be fixed to the bottle while the electrical components can be removed and serviced. The figure below shows the whole design of the electrical housing.

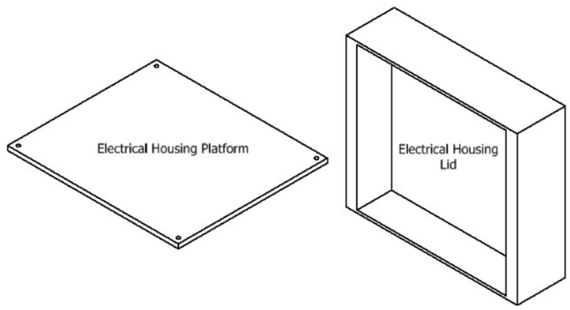

Fig. 8. Electronic housing Design

D. *Ultrasonic Sensor Lid Mount Design*

To get accurate readings the position of the ultrasonic sensor is vital. If the sensor is off by a few millimeters, the results produced by the sensor would be unreliable. To solve this issue, we designed a mount to rigidly hold the sensor in place. When the lid is fully tightened the sensor returns to the same place in relation to the bottle for consistent sensor readings. To approximate the general position that the sensor needed to be in, a rough markup with a cardboard sheet was produced. With the position of the sensor cut out of the cardboard with relation to the bottles lid made it easy to translate these measurements into a CAD model. The ultrasonic sensor sits on a raised platform from the bottom of the lid. The offset is to get the sensor below a lip in the bottoms design.

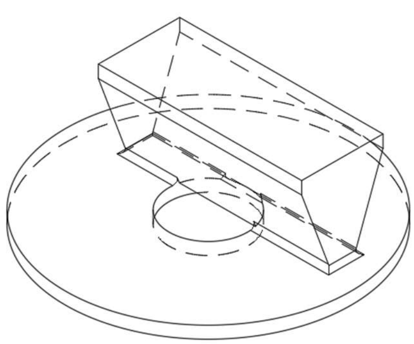

Fig. 9. Ultrasonic Sensor Lid Mount

IV. OPTICAL SYSTEM DESIGN

In this section we will show the theory and process behind the design of the single wavelength spectrometer. The spectrometer is used to measure the absorption of light through a sample. Each section will discuss the theory and selection of each component.

A. LED

The LED is one of the primary components for the spectrometer sub system. The design of the spectrometer is centered around testing glucose concentration in sports beverages. The desired wavelength for testing glucose concentrations is 580 nm. Thorlabs provides a 591nm LED that has a bandwidth that covers the 580 nm range. This is what we selected for our project. Important parameters for the LED include the optical output power of the LED which is determined by the series resistance. According to the data sheet the ideal operating voltage of the LED is 2.2V at 20 mA of current. To determine series resistance, we used Kirchoff's voltage law.

$$R_s = \frac{V_{ESP} - V_{LED}}{I_{LED}} \tag{1}$$

$$R_s = \frac{3.3\,V - 2.2V}{20\,mA} = 55\,\Omega \tag{2}$$

The equation above shows that the LED needs a series resistance of 55 ohms to output 2mW of power. 2mW of power is the maximum power output of this LED according to the data sheet provided by Thorlabs. We can calculate the theoretical output power using the following equation

$$P = I^2 R_s = (20\,x10^{-3}A)^2(55\Omega) = 2.2\,mW \tag{3}$$

B. Collimating lens

The collimating lens is responsible for taking the diverging rays from the LED and making them parallel along the optical axis. If the light were allowed to continue diverging the total optical power will be significantly reduced therefore collimation of the light is paramount to signal received by the light sensor. We can use the lens maker equation to solve the position of the collimating lens in relation to the object. The collimating lens selected is a Newport biconvex lens with a focal length of 38.1 mm. To represent collimated light, the focusing of the light should be equal to infinity.

$$\frac{1}{l1} + \frac{1}{l2} = \frac{1}{f} \tag{4}$$

In the equation above we see that l1 and l2 are the distance between the object and lens and the lens to the focal point respectively. We know that the focal length of the lens is 38.1 mm, and we know that the distance from the lens to the focal point needs to be equal to infinity. So, if we plug that in, we can find how far away the lens needs to be placed from the light source to collimate the light.

$$\frac{1}{l1} + \frac{1}{\infty} = \frac{1}{38.1} \tag{5}$$

From the equation we find that the lens should be placed exactly 38.1 mm away from the aperture to collimate the beam.

C. Focusing lens

To focus on the light that travels through the sample we must place the lens 38.1 mm away from the light sensor. A full depiction of how light will travel through the system can be found below.

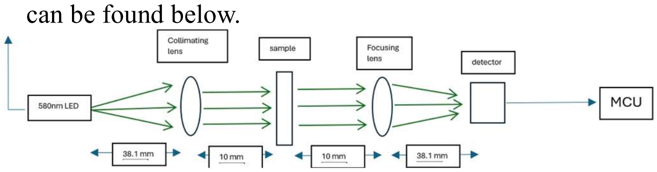

Fig. 10. Design Schematic for Spectrometer Sub system

D. Light Sensor

The light sensor we chose is the tsl2591 ambient lux sensor from Adafruit. We chose this light sensor, because it has i2c communication, variable gain and integration time, and will give us readings we can use to measure absorption of a single wavelength. The most important parameter for the sensor is the signal to noise ratio. To test SNR, we must take an ambient light reading and a blank reading for intensity of LED. We can then calculate the SNR using the following equation.

$$SNR = -20 \log \left(\frac{Ambient}{Intensity} \right) \tag{6}$$

Ambient reading is equivalent to the noise and the intensity reading is our signal. Furthermore, we can calculate the theoretical SNR of the sensor using the data sheet and expected output power of the LED. To calculate the theoretical signal to noise ratio we need the noise equivalent power and the optical output power. See below for the theoretical SNR calculations.

$$SNR_{theoretical} = -20 \log \frac{NEP}{Intensity} \tag{7}$$

$$SNR_{theoretica} = -20 \log \frac{4.98 \mu W}{2mW} = 52dB \tag{8}$$

E. DNSA Testing

The 3,5-dinitro-salicylic acid (DNSA) test is a color changing test. A reaction with the DNSA reagent and a reducing sugar produces a range of colors within yellow and a dark amber. The DNSA reaction was chosen, due to how quickly we were able to implement the tests in senior design 2. There are other methods that use enzymes to form a color changing reaction, but with limited time to meet deadlines we chose this method. The process occurs when the DNSA reagent (made from mixing 3,5-dinitro-salicylic acid and NaOH) is mixed with a dilute solution containing a reducing sugar. Once heated the color of the solution will become darker depending on the concentration of sugar. We can then run the sample in our single wavelength spectrometer to measure the absorption of light using the following equation.

$$A = -\log \left(\frac{Sample\ intensity}{blank\ intensity} \right) \tag{9}$$

In the equation above we find that A is the absorption of light, which is a ratio of the light readings through the sample and the average intensity of light when there is no sample. We can then take the value of absorption (any number between 0 and 1) and plug it into Beer's law.

$$A = \varepsilon bc \tag{10}$$

Beer's law states that the absorption is equal to ε the molar absorption coefficient multiplied by the light path through the cuvette(b) multiplied by the concentration(c). The value for the molar absorption coefficient of glucose at 580 nm is approximately close to zero. We ran calibration tests using stock solutions to find the absorption coefficient and found it to be $1.6 mM/cm^3$. With that information we also know that our cuvette is 11.8 mm in width. Rearranging Beer's law we can calculate the concentration of glucose in beverages.

$$c = \frac{A}{\varepsilon b} \tag{11}$$

The equation will give us the molar concentration of glucose which we will then convert to grams by multiplying it by the molar mass of glucose which is 180.156. The last thing we must do is multiply that number by the dilution factor to get the concentration in grams per milliliter.

V. Electrical Design

A. *Microcontroller*

The ESPRESSIF ESP-32 is a small board filled with features. Including Bluetooth, Wi-Fi, and 2 processing cores at 240MHZ should we need it. In an embedded system one of the most important resources is the amount of SRAM available. The ESP-32 has 520 KB onboard that should be plenty for our purposes.

Our needs require the ability to interact with a lot of peripherals. This board has UART, SPI, I2C, ADC, LED PWM, and plentiful GPIO pins. The ESP-32 is a standout in that it has built-in capacitive touch sensors, which is perfect for our application. With all these peripherals in mind the ESP-32 more than satisfies all our requirements in that department. This board also has low power draw which is another necessity. It has an average operating current of 80mA with an operating voltage of 3.0 to 3.6 volts, which fits perfectly within the range of our voltage regulator.

B. Touch Sensor

As discussed above, the ESP32 MCU is equipped with built in capacitive touch sensors. This is because a certain number of GPIO pins are equipped with this capability. For this case only one pin is necessary. This approach offers many advantages. First, as it is built into the MCU there is no need for extra components to take up space. Also, this could potentially reduce points of failure in the overall system. Being that there are less components and reduced overall complexity. To implement this design a programming library is provided to set the GPIO pins to their touch sensor functionality. Once the board is programmed the only other requirement is a wire attached to the GPIO pin. Now, attaching a wire to the bottle is not exactly the most elegant design. To expand the touch surface, some other conductive material is needed to attach to the GPIO pin. Copper or aluminum tape is a likely option. Although, attaching the GPIO pin to the conductive tape could serve as a point of failure. The final benefit to this design is the plentiful documentation provided for the programming library.

C. *LED Alert System*

For ease of use this system will have an LED notification system to let the user know when they need to take a drink of water based on the last time they drank. Currently when the user touches the bottle the LED turns off. We use a ten-second timer that activates the LED once the interrupt flag is raised, simulating a longer interval, such as two hours between drinking water or other liquids.

D. *Water Level Sensor*

To track the amount of water the user has drank we need to implement a sensor to determine the amount of water left in the water bottle, we considered many choices such as a scale, capacitive water sensors and ultrasonic. We ruled out capacitive and scale technologies due to the fact that we do not have experience with capacitive sensors or that the scale will not take into consideration some liquid types such as coke and water might have different weights therefore skewing out data for water levels.

Therefore, ultrasonic is our choice technology. Within this category there is one particular ultrasonic sensor that fits our needs and that was the A02YYUW ultrasonic sensor. This sensor has a very small blind zone compared to the other we researched as well as it is waterproof at a rating of IP67 which is perfect to implement into a potentially wet environment. With an operating voltage of 3.3-5v it is well within our means for electrical design.

The data that will be sent to the app will be a linear function which converts the CM of water to a percentage of amount of water left in the bottle from a maximum we set depending on the blind zone of the A02YYUW.

E. Main Board Schematic

Our main board is controlled by an ESP32 that handles the task of gathering data from all the peripherals as well as sending data to the app. For design ease only the ESP circuits and peripherals such as the LEDs, Ultrasonic sensor and the CP2102 are integrated onto the main board.

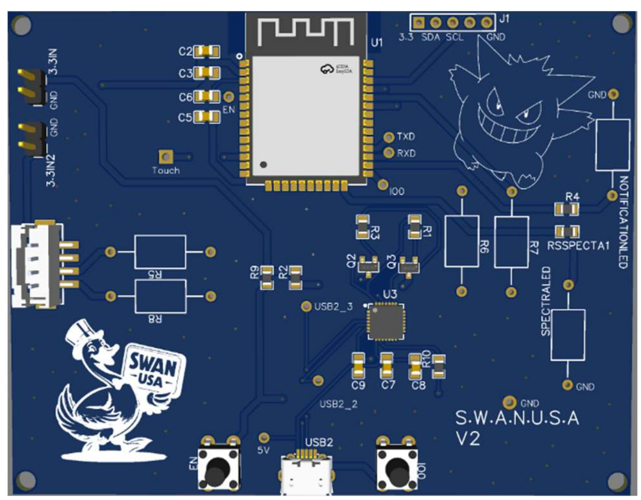

Fig. 11. Main Board PCB Layout

There are two separate power lines, one that powers solely the ESP and the other power line powers the other components. For programing we included the use of micro-USB and buttons to change upload states.

All the breakout boards and the LEDs since not connected straight onto the Main board will be connected using pin headers which are mounted onto the edges of the PCB.

F. *PSU Board Schematic*

SWANUSA's power supply board features 2 LDO's. LDO's were chosen because switching regulators are very susceptible to noise and incorrect PCB design compared to linear technologies.

The LDO of choice is the ADP3338, this is an Ultra Low Dropout regulator which has a dropout voltage of about 125 mV under standard conditions and can supply up to 1A of current to its load. Since this is an Ultra LDO regulator we can use one battery to drop a nominal voltage of 3.7V to 3.3V for an efficiency of 89% which is in the realm of most switching regulators

As mentioned in the section above two powerlines come out from this board design which splits the load of the regulators by having one regulator power the ESP32 and the peripherals respectively, header pins are used to connect this breakout board to the main PCB.

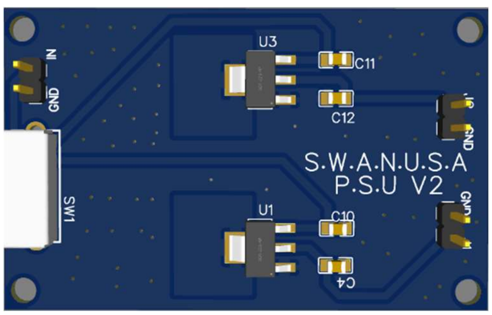

Fig. 12. PSU Board Layout

G. *Light Sensor PCB*

The light sensor of choice is the TSL2591, this breakout board was designed using the recommended schematic from the data sheet of the part. Since this sensor used I2C protocol we needed to implement the use of pull-up resistors as seen on the schematic. Like all the breakout boards and Main PCB in this project were using header pins to connect this breakout board to the main board for communication to the ESP32.

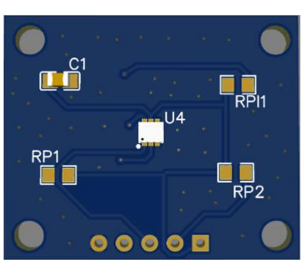

Fig. 13. TSL2591 Layout Schematic

VI. Software Design

This section will outline the full stack development of the mobile app. We will cover the software block diagram, front of house software choice and design, back-end data relation and processing, and user authentication and login. Below is the Software Block Diagram that shows how the software will interact with our selected microcontroller.

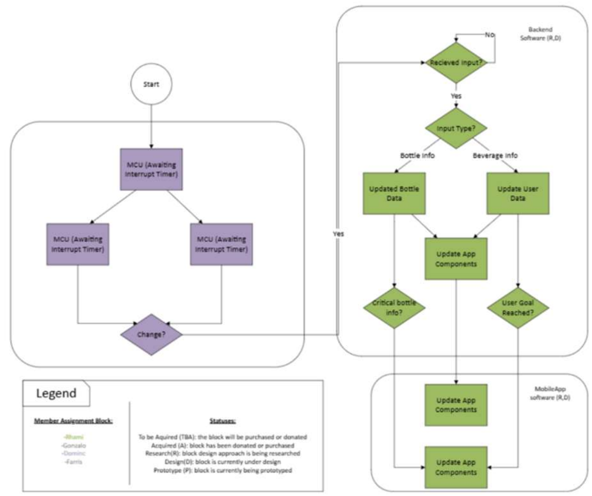

Fig. 14. Software Block Diagram

In the diagram above we can note that software awaits a change in data that is sent over Bluetooth. If there is no change it loops back and waits for a change to be detected. If a change is detected, then it will sort the information between bottle info and beverage info. Bottle info relates to ultrasonic sensor readings and LED alerts, while beverage info is related to data being sent from the spectrometer. From there the software will update the information in the back end of the app and display critical information to users.

A. *User Login and Authentication*

This section will show how the user will set up their account and what that will look like both in the front and back-end software. Furthermore, we will also showcase how a returning user will sign back into the app. User authentication is also an important aspect to ensure stored data from the ESP is displayed to the appropriate user. The first figure will show the user interface for creating an account. Lastly figure 2 will showcase the user login for returning users.

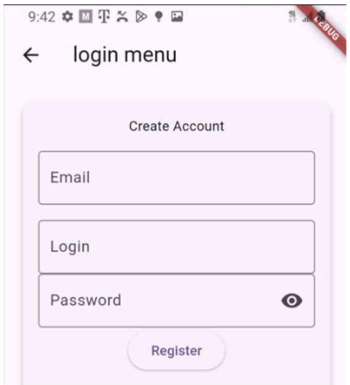

Fig. 15. Screenshot of account creation

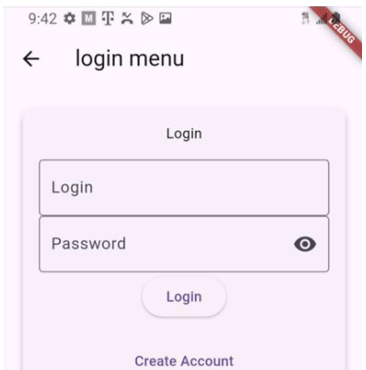

Fig. 16. Screenshot of user login

B. *Front-End App Development*

For front end development we chose to use flutter. Flutter enables us to customize the UI and is capable of cross-platform app development. In this section we will cover how we used flutter to design the user interface and what our final app will look like.

To begin we used flutter with an object-oriented programming (OOP) approach to segment major components of the User interface (UI) such as Main page, Bluetooth, and the Login/register UIs. With this we also used many Flutter packages such as Flutter Blue Plus to help with Bluetooth functionality, AWS packages for the connection to the backend, and Flutter secure storage to support long term storage of user variables. We also leveraged OOP concepts to help create a class to pass user information and variables amongst all the UI classes. Above you will see screenshots of the UI specifically the main page and the Bluetooth connection page, the login/register was shown in section A.

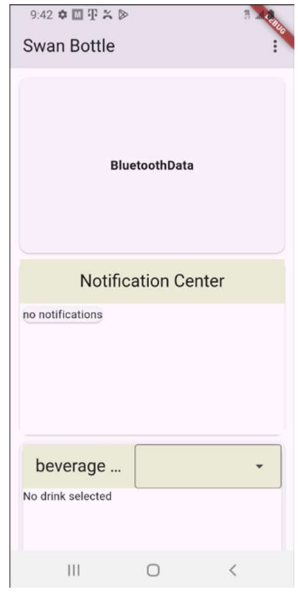

Fig. 17. Screenshot of user interface

C. *Back-End App Development*

The back-end development is split into two parts. Firstly, we used DynamoDB for the database and amplify for user authentication. Both of these AWS services proved to be vital in the development of the app. Bellow we will outline the usage of both Amplify and DynamoDB in more detail.

To begin Amplify was used to help do the brunt of the user authentication. From a high level view the user would in the app either select to login/logout or register a user, and the user will be redirected to that UI to put in the proper credentials. If the user is trying to login, then the user will put in a username and password and then with the use of the AWS packages on flutter will then send a login request which will either be accepted and create a session or denied. If The user selects logout, then this session will be closed, and their login info will be deleted from the appropriate class. If the user registers it operates similar to login, but it also asks for an email and an email verification code which is handled by AWS Refer to section A for more detail on the login.

The DynamoDB was used to store beverage information and user information for later usage. At the current feasible scope of this project, we are currently just storing user information to then implement Widgets for later updates using this data. We are also using DynamoDB to store beverage data and to load tables to show the user what they are drinking and its nutritional value.

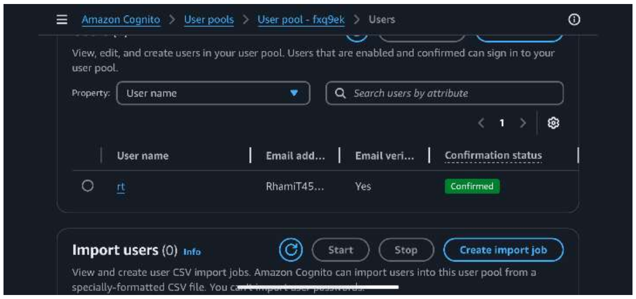

Fig. 18. Screenshot of registered users

D. *Embedded Systems*

The embedded system section will primarily discuss the codes functionality for the peripherals. This includes an ultrasonic sensor, LED alert system, touch sensor, and spectrometer. All programing of the embedded systems was done using the Arduino IDE software.

It is programmed so that the system only takes readings from the sensors once the touch sensor has registered a touch. The system is dormant while no one touches the touch sensor. Once touch is registered the MCU will take a reading from the light and ultrasonic sensor then transmit those readings over Bluetooth.

The ultrasonic sensor communicates with the MCU via UART. For this sensor the UART data is in 4 frames. Two data frames, a header frame, and a checksum. Due to this in the code we must do some operations to extract the data and convert it to centimeters. Once the data is in centimeters, we will plug that value into a function to calculate a percentage of how full the bottle is.

The spectrometer is programmed to take three separate readings. The three readings will be the average of 10 lux values measured by the tsl2591. To do this we have created two separate functions. One function serves to measure the average signal noise when the LED is turned off and the other function will take the average reading of the lux sensor for intensity and transmission through the sample. The 10 measurements are stored in three separate integer arrays labeled for the measurements. There is another function for averaging the data points that we can call after running the reading function. Once the data has been collected the program will calculate the SNR and absorption based on the readings from the spectrometer. From that it also calculates the concentration in mols/L and converts it to grams/ml.

VII. CONCLUSION

In conclusion we believe our project takes the next step in technological integration in daily consumer products. Our interdisciplinary team has worked hard to seamlessly integrate optics, full stack development, and electrical engineering principles to bring SWAN-USA to life. In summary the electrical engineers designed our PCB and peripherals to meet specified goals and achieve a portable smart water bottle. Our optical engineer honed principles of light to provide unique never seen (in a water bottle) analytics. Lastly, our computer science major tackled programming a full stack development that takes an entire group to accomplish solo.

ACKNOWLEDGEMENT

We wish to acknowledge the support and mentorship received during the design and implementation of the Smart Water Bottle with Nutrition Concentration Measurement from Dr. Arthur Weeks for guiding us through optical design and implementation. Dr. Stephen Kuebler for acquiring the DNSA chemical and for letting us utilize his research lab for testing. Dr. Stephen Eikenberry for mentoring us with the spectrometer design. Thank you for making this project possible!

REFERENCE

1. PerkinElmer. Determination of Sugar as Glucose in a Soft Drink Using the LAMBDA PDA UV/Vis Spectrophotometer. 2015-2016.

2. D. Knight, "Introduction to Linear Voltage Regulators," 13 Februrary 2016. [Online]. Available: https://www.digikey.com/en/maker/tutorials/2016/introduction-to-linear-voltage-regulators. [Accessed 2024].

3. Amazon Web Services (AWS), "Introduction to Amazon DynamoDB," Amazon Web Services Documentation. [Online]. Available: https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/Introduction.html. [Accessed: Nov. 24, 2024].

4. Rima, Jamil, and Karin Assakar. "(PDF) a New Spectrophometric Method for the Analysis of Fructose, Glucose and Sucrose, Using 2-Thiobarbituric Acid and Zero-Valent Iron Powder (ZVIP)." Research Gate, Feb. 2019, www.researchgate.net/publication/331176548_A_New_Spectrophometric_Method_For_The_Analysis_Of_Fructose_Glucose_And_Sucrose_Using_2-Thiobarbituric_Acid_And_Zero-Valent_Iron_Powder_ZVIP.